REMARKS

The enclosed is responsive to the Examiner's Office Action mailed on September 29, 2006. At the time the Examiner mailed the Office Action claims 1-14 and 16-23 were pending. By way of the present response the Applicants have: 1) amended claims 1, 16-18, 20, and 22; 2) added no new claims; and 3) canceled claims 7 and 21. As such, claims 1-6, 8-14, 16-20, and 22-23 are now pending. The Applicants respectfully request reconsideration of the present application and the allowance of all claims now presented.

Claim Objections

The Office Action requested that the phrase "thread partitionable" be replaced with "thread-partitionable" in all of the claims. However, Applicants respectfully submit that the requested phrase is not what Applicants used in the specification and feel that such a change would lead to less clarity in the claims and have not made the requested change.

Claim 1 was objected to for two informalities. Applicants have amended this claim as suggested in a non-narrowing manner.

Claim 16 was objected to as being dependent upon a canceled claim 15. Applicants have amended this claim as suggested to make the claim dependent upon claim 14.

Claim 20 was objected to because it did not have a period. Applicants have amended this claim as suggested to correct this typographical error.

Claim Rejections

35 U.S.C. 112 Rejections

The Office Action rejected claim 17 under 35 U.S.C. 112, second paragraph, as being indefinite. Specifically, that the limitation "the program instruction" in line 6 lacked antecedent basis. Applicants have amended claim 17 in a non-narrowing manner accordingly.

35 U.S.C. 102(e) Rejections

The Office Action rejected claims 1-2, 10 and 18-19 under 35 U.S.C. 102(e) as being anticipated by Eickemeyer, U.S. Patent 6,931,639 (hereinafter "Eickemeyer").

Applicants amended claims 1 and 18 to include the limitations of claims 7 and 21 respectively. Claims 7 and 21 were rejected under The Examiner rejected claims 7- under 35 U.S.C. 103(a) as being unpatentable over Eickemeyer in view of Kalafatis, et al. U.S. Patent 6,535,905 (hereinafter "Kalafatis"). Applicants will address claims 1 and 18 under this 103(a) rejection below.

Claims 2, 10, and 19 are dependent upon claims 1 and 18 respectively and are allowable for at least the same reasons.

35 U.S.C. 103(a) Rejections

The Office Action rejected claims 7-9, 13-14, 16-17, and 21-23 under 35 U.S.C. 103(a) as being unpatentable over Eickemeyer in view of Kalafatis, et al. U.S. Patent 6,535,905 (hereinafter "Kalafatis").

Eickemeyer describes "implementing a variable-partitioned queue for simultaneous multithreaded processors." (Eickemeyer, Abstract.) Eickemeyer describes three queues "the LRQ [load reorder queue] 228, SRQ [store reorder queue] 230, and GCT [global completition table] 232 can be variably partitioned between threads." (Eickemeyer, Col. 3, lines 48-50.) Eickemeyer does not describe that the rename pools 234 is "variably partitioned between threads." Presumably, these queues (except for the rename pools 234) are described in more detail with respect to the exemplary queue structures figures 3 and 4, however, Eickemeyer never explicitly says what these exemplary queues pertain to.

Kalafatis describes "thread switching operation within a multithreaded processor" in certain situations. (See Kalafatis, Abstract.) For example, Kalafatis describes "that [when] the allocator [of a microprocessor] determines that insufficient resources ... are available for instructions (i.e., microinstructions) for a particular thread received from the queue ... the allocator 76 asserts a stall signal ...On the assertion of such a stall signal 82 for a particular thread, it may be desirable to perform a thread switching operation." (Kalafatis, Col. 13, lines 32-46.) Kalaftis is not describing relinquishing portions of thread partionable resources but merely that if

there are insufficient resources available for a particular thread it may be desirable switch to a different thread.

Kalafatis further describes "[i]n a multithreaded processor, where a page miss occurs for an instruction stream of a current thread, it may be advantageous to perform a thread switching operation so as to allow an alternative thread to utilize the latency introduced by the page walk operation. ... [A] determination is made as to whether a predetermined minimum quantity of instruction information (e.g., a predetermined minimum number of chunks) for an alternative thread (e.g., thread 1) are pending and available for dispatch from the logical partition 124 of the instruction streaming buffer 106." (Kalafatis, Col. 18, line 62 through Col. 19, line 53.) Kalaftis is not describing relinquishing portions of thread partionable resources but determining if a minimum number of instruction information is available for an alternative thread to process.

The combination of Eickemeyer and Kalafatis does not describe what Applicants' claim 1 requires. Specifically, the combination does not describe:

> a plurality of thread partitionable resources that are each partitionable between a plurality of threads including a first thread and at least one other thread, wherein said plurality of thread partitionable resources comprise:
>
> an instruction queue, and
>
> a register pool;
>
> a plurality of shared resources shared by the plurality of threads including the first thread and at least one other thread; and
>
> logic to receive a program instruction from a first thread directing said processor to suspend execution of said first thread, and in response to said program instruction, to cause the processor to suspend execution of the first thread and to relinquish portions

of said plurality of thread partitionable resources associated with the first thread for use by at least one other thread of said plurality of threads.

Neither Eickemeyer nor Kalafatis describe that thread switching causes the relinquishing a partition of an instruction queue or a plurality of registers from a register pool as required by the claim. Furthermore, Eickemeyer explicitly leaves out the instruction buffer, instruction cache, and rename pools from being "variably partitioned between threads" and therefore provides no motivation for relinquishing a partition of an instruction queue or a plurality of registers from a register pool.

Accordingly, Applicants respectfully submit that claim 1 is not described by the combination of Eickemeyer and Kalafatis. Claims 2-6 and 8-12 are dependent upon claim 1 and are allowable for at least the same reason.

The combination of Eickemeyer and Kalafatis does not describe what Applicants' claim 13 requires. Specifically, the combination does not describe:

receiving a first opcode in a first thread of execution;
suspending said first thread for a selected amount of time in response to said first opcode;
relinquishing a plurality of thread partitionable resources in response to said first opcode, wherein relinquishing said plurality of thread partitionable resources comprises:
relinquishing a partition of an instruction queue; and
relinquishing a plurality of registers from a register pool.

Neither Eickemeyer nor Kalafatis describe that thread switching causes the relinquishing a partition of an instruction queue or a plurality of registers from a register pool as required by the claim. Furthermore, Eickemeyer explicitly leaves out the instruction buffer, instruction cache, and rename pools from being "variably partitioned

between threads" and therefore provides no motivation for relinquishing a partition of an instruction queue or a plurality of registers from a register pool.

Accordingly, Applicants respectfully submit that claim 1 3is not described by the combination of Eickemeyer and Kalafatis. Claims 14 and 15-17 are dependent upon claim 13 and are allowable for at least the same reason.

The combination of Eickemeyer and Kalafatis does not describe what Applicants' claim 18 requires. Specifically, the combination does not describe:

> a memory to store a plurality of program threads, including a first thread and a second thread, said first thread including a first instruction;
> a processor coupled to said memory being separate from said processor, said processor including a plurality of thread partitionable resources and a plurality of shared resources, said processor to execute instructions from said memory, said processor, in response to execution of said first instruction to suspend said first thread and to relinquish portions of said plurality of thread partitionable resources, wherein said plurality of thread partitionable resources comprise:
> an instruction queue, and
> a register pool.

Neither Eickemeyer nor Kalafatis describe that thread switching causes the relinquishing a partition of an instruction queue or a plurality of registers from a register pool as required by the claim. Furthermore, Eickemeyer explicitly leaves out the instruction buffer, instruction cache, and rename pools from being "variably partitioned between threads" and therefore provides no motivation for relinquishing a partition of an instruction queue or a plurality of registers from a register pool.

Accordingly, Applicants respectfully submit that claim 18 is not described by the combination of Eickemeyer and Kalafatis. Claims 19-20 and 22-23 are dependent upon claim 13 and are allowable for at least the same reason.

In light of the comments above, the Applicant respectfully requests the allowance of all claims.

CONCLUSION

Applicant respectfully submits that all rejections have been overcome and that all pending claims are in condition for allowance.

If there are any additional charges, please charge them to our Deposit Account Number 02-2666. If a telephone conference would facilitate the prosecution of this application, the Examiner is invited to contact Thomas C. Webster at (408) 720-8300.

Respectfully Submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 2/28/07

Thomas C. Webster
Reg. No.: 46,154

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300